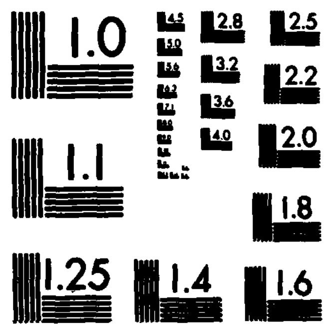
AD-A144 484 GAZE CONTROL DURING HORIZONTAL AND VERTICAL TARGET TRACKING(U) CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF BIOMEDICAL ENGINEERING A T BAHILL MAR 84 1/1

UNCLASSIFIED AFOSR-TR-84-0698 AFOSR-83-0137 F/G 5/10 NL

END



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 84-0698	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GAZE CONTROL DURING HORIZONTAL AND VERTICAL TARGET TRACKING		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dr. A. Terry Bahill		8. CONTRACT OR GRANT NUMBER(s) AFOSR-83-0137
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of ~~Electrical & Computer~~ B... Engineering, Carnegie Mellon University, Pittsburg, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2313/D9 61102F
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NL Bolling AFB, DC 20332		12. REPORT DATE March 1984
		13. NUMBER OF PAGES 16
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

DTIC ELECTE AUG 20 1984 S D E

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Honeywell Oculometer, Eye Movements, Gaze Control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Honeywell oculometer has a noise level of about 0.1 deg; eye tracking is noisier than head tracking; vertical eye tracking is noisier than horizontal eye tracking. It has about 25% crosstalk of the horizontal channel into the vertical channel. It has an 84 ms time delay. It is not effective at detecting and rejecting eye blinks; typical eye blink artifacts last 50 to 200 ms. The human tracks best when tracking with eyes alone. Although tracking with head and eyes should be more natural, the human does worse when he uses his head. Head only tracking is the worst of the three conditions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AFOSR-TR- 84-0698

Final Technical Report
AFOSR-83-0137
March 1984

GAZE CONTROL DURING HORIZONTAL AND VERTICAL TARGET TRACKING

Dept of Biomedical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Dr. A. Terry Bahill

Accession For		
NTIS GRA&I		X
DTIC TAB		
Unannounced		
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		

Controlling Office: Air Force Office of Scientific Research/NL
Bolling Air Force Base, DC 20332

INTRODUCTION

Can a human track best with eyes alone, with head alone, or with coordinated head and eye movements? Although coordinated head and eye movements seem most natural, I have found, quite surprisingly, that humans track better if they use only their eyes.

How could each of these situations arise? It seems that we use coordinated head and eye movements in our everyday life. And indeed studies have shown that professional athletes do track targets with both head and eyes (Bahill and LaRitz, 1983). On the other hand, eye only tracking is common in scientific laboratories where the head is held stationary with a bite bar or a head rest. There are two reasons for fixing the head in these studies. First, if the head is fixed, then only eye movements have to be measured in order to determine the direction of gaze. And it is easier to measure only eye movements rather than head and eye movements. Second, in studying physiological systems, it is common to simplify the task by minimizing the number of systems involved. So, when head movements are eliminated the system reduces to eye movements only. On the other hand, head only tracking occurs in certain machines, such as the Army's Apache helicopter, where the head is used to direct guns or forward looking infrared radar. Only head movements are measured. To make sure that the head is pointed at the target, a circle is painted on the visor in front of the pilot's eye, and the pilot superimposes the circle on the target. This insures that the eyes are looking straight ahead, and, therefore, that the head position indicates the direction of gaze. This paper shows examples of each of these three tracking strategies (eyes alone, head alone, or coordinated head and eye movements) and it shows which is the most successful.

METHODOLOGY

Head and eye movements were measured with the Honeywell Helmet Mounted Oculometer in the Helmet Mounted Oculometer Facility of the Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base. This system is similar to the one described by Merchant, et al. (1974). Head movements were measured with a standard Polhemus head tracking system. And Eye movements were measured with a TV system. A small TV camera was mounted on the helmet visor and it looked an image of the eye that was reflected off of a half-silvered mirror deposited on the visor in front of the right eye.

Further details of this oculometer, such as the phrasing diagrams for the HMOS data paths, can be found in the Operation and Maintenance Manuals for the Helmet Mounted Oculometer Subsystem prepared by the Avionics Division of Honeywell Inc. These manuals are proprietary, so a discussion of their contents would be inappropriate for this report.

On the average the Honeywell oculometer computes a new angle every 16.7 msec (with a standard deviation of 1.8 msec). This limits it to a 30 Hz bandwidth. An FFT analysis revealed that the smooth pursuit eye movements of this report only had information below 1 Hz. Thus the oculometer did not limit the frequency content of these smooth pursuit eye movements. However an FFT analysis of saccadic eye movements has shown information up to 40 Hz (Bahill, et al., 1982). So the oculometer does limit the information contained in saccades.

EYE BLINKS

Eye blinks produce artifacts in the output of the Honeywell eye tracker. Figure 1 shows what two particular eye blink artifacts look like. The spikes at the beginning and end of the record are due to eye blinks. The first one produces a bidirectional artifact. First there is a 100 millisecond (msec) downward deflection, and 301 msec later there is a 184 msec upward deflection. The blink at the end of the record produced a downward spike 134 msec in duration. Converting milliseconds to data points yields spike durations of 6, 8 and 11 data points.

The reason I am convinced that these spikes are due to eye blinks is that they occurred just before and just after target tracking. Good subjects concentrate hard and try to suppress eye blinks during tracking, and therefore, you usually see eye blinks just before and just after target tracking. Furthermore, these spikes look like eye blinks that I have observed with several other instrumentation systems. In addition, they are too fast for back-to-back saccades.

Figure 3 shows blink artifacts in the vertical channel. Note that the blink at the beginning of the record produced spikes in opposite directions in the horizontal and vertical channels. Whereas, the blink at the end of the record produced spikes in the same direction. Another blink is shown at the 9 second point in Figure 2. It was only 50 msec long. (It actually is in the horizontal channel too, but it does not show up in the plot, because I skip points when plotting out data.)

NOISE

I have computed the mean squared error of the position traces when the human was fixating on a stationary target. (Sections similar to the start of Figures 1 to 5, except longer and free of blinks or other artifacts.) The target was stationary so I call any apparent eye movement noise. It is primarily instrumentation noise, because biological noise is only 0.001 deg^2 (Hsu, et al., 1976; Eizenman, et al., 1984). For the Honeywell oculometer, the noise in the horizontal eye channel was 0.11 deg^2, the noise in the vertical eye channel was 0.12 deg^2, the noise in the horizontal head channel was 0.05 deg^2, and the noise in the vertical head channel was 0.07 deg^2. What do these numbers mean? Well first off, it means that the head tracker is better than the eye tracker. Second, it means that horizontal eye tracking is better than vertical eye tracking. It is not likely that the noise in the vertical eye channel can be reduced, because it is probably do to the limited resolution of the 256 line/frame TV system. Third, to give the numbers some perspective, I have computed similar noise terms for horizontal eye position measured with other systems. Electro-oculography (EOG) has a typical noise level of 0.3 deg^2 and our photoelectric data (e.g. Figure 6) has a typical noise level of 0.01 deg^2. This means that the Honeywell oculometer has lower noise than EOG. And it approaches the noise level of the photoelectric technique.

GAZE CALCULATIONS

Figure 1 shows the comparison of two different methods of calculating gaze angle. I have taken HMOF's head angle and HMOF's eye angle and from these I computed the gaze angle. Then I superimposed this trace on HMOF's calculation of gaze angle. The two calculations of gaze angle are different. The differences are probably due to translations, either translations of the helmet or translations of the head inside the helmet. HMOF's calculation of gaze included the effects of translation; mine did not.

TARGETS

The target was a small dot presented on an X-Y monitor 53 cm in front of the subject. It was driven by sinusoidal voltages five degrees in amplitude. Therefore, it moved either horizontally or around a circle 10 degrees in diameter. Frequencies of 0.25 and 0.1 Hz were used.

RESULTS

Can the human track better with eyes alone, with head alone or with coordinated head and eye movements? Quite surprisingly the answer is that the human can track best with eye movements only, as is shown in Figs. 2, 3 and 4.

For Figure 2 the subject held his head stationary and tracked the target with only his eyes. The eyes followed the target well and there were no measured head movements. When the target started, there was a 234 msec delay and then a 4 degree saccadic eye movement in both the horizontal and vertical channels. The rest of the record contains smooth pursuit eye movements with occasional small position correcting saccades. The lowest trace shows horizontal gaze superimposed on the horizontal target position; the subject tracked well. The mean squared error (mse) between the subject's gaze and the target position was 0.35 deg^2.

In Figure 3 the subject tracked the target with his head and eyes. Once again tracking began with 4 degree eye saccades. The subsequent tracking was a combination of smooth eye and head movements. Both the head and the eye seemed to be in phase with the target. The sum of the head and the eye accurately tracked the target. The mse between target and gaze positions was 0.50 deg^2. The spikes at the beginning and end of the records are eye blinks.

In Figure 3 vertical head movements were larger than horizontal head movements, but this was not a consistent trend in the rest of the data.

In Figure 4 the subject tracked with head movements only. However, we did not glue his eyes in his orbits, so we had to trust him not to move his eyes. For the most part, he did this. However, when the target just started moving, he could not resist. He started out with fast saccadic eye movements, followed immediately by vestibulo-ocular eye movements in the opposite direction. These vestibulo-ocular eye movements were a consequence of the head movements. This type of head and eye coordination is perfectly normal. Bizzi and Barnes (Morasso, et al., 1973; Barnes, 1980) have reported that this is the most common type of movement when both head and eyes are free. The reason for pointing it out, is that the subject himself was probably not aware that he was doing this. The mse between target and gaze positions was 1.09 deg^2.

There is an artifact in the yaw record of Figure 4; the small humps at 5.7, 10.0, and 14.3 sec should be downward not upward. Fortunately, this artifact did not effect the computation of horizontal gaze.

CROSSTALK OF THE HONEYWELL OCULOMETER

Figure 5 shows data of a subject tracking a target moving horizontally. There was no vertical target movement. Therefore, we should expect no vertical eye movements. However, we see vertical eye movements that are about 25% the size of the horizontal eye movements. I can envision three possible reasons for this crosstalk. (1) The target may not have been moving horizontally; it might have been tilted. The experimentors who collected the data did not say that this was the case. (2) The human could be moving in a figure eight pattern, instead of moving back and forth horizontally. Subjects in my laboratory do not track in a figure eight pattern. (3) Therefore, I believe that the oculometer has 25% crosstalk of the horizontal channel into the vertical channel. This is probably due to inaccuracies in the calibration or linearization files.

TIME DELAY OF THE OCULOMETER

There is a time delay in all three tracking conditions. I have used the crosscorrelation function to estimate the approximate time delay for these data. I caution that the time of peak crosscorrelation is not a perfect measure of the time delay of a system (Bahill and McDonald, 1983), but it is still a reasonable one. Using this function to estimate the time delay yields values of 184 msec for head alone tracking, 150 msec for eyes and head tracking, and 83 msec for eyes alone tracking. (These time delays were similar for our second subject

146, 92 and 63 msec.) This also shows that the human tracks better if he does not use his head. However, a normal human can track predictable targets with zero latency (Bahill and McDonald, 1983). Figures 6 and 7 show zero-latency tracking of sinusoidal targets. These figures were derived from different subjects using different eye movement measurement equipment. Figure 6 shows data from an experienced subject and Figure 7 shows data from a naive subject. The mean squared errors between target and eye positions were 0.23 and 0.02 deg^2. It is obvious that they were tracking with no latency, in contrast the data of Figs. 2, 3, and 4 that show a large latency. Therefore we suggest that the Honeywell eye tracker has a time delay of about 83 ms.

DISCUSSION

The human can track better using only eye movements than using only head movements, or using head and eye movements. One possible explanation for this is that when the head is used for tracking the vestibulo-ocular reflex must be suppressed. It may be easier to suppress head movements than to supress the vestibulo-ocular reflex.

DIRECTIONS FOR FUTURE RESEARCH

These data have suggested that the Honeywell oculometer has a time delay of about 84 msec, which is larger than the expected 17 msec. There are two ways to further substantiate this time delay. First many Air Force subjects could be run at WPAFB and the data could be analyzed on my computer graphics system. Or, second, I could bring a few subjects, who I know can track with zero latency, to WPAFB. We would run them and I would analyze the data with my computer graphics system. These further tests would help to establish whether the apparent time delay is due to the Honeywell oculometer, or to the two particular subjects that were used in the experiments described in this report.

Because there is a large time delay in the Honeywell oculometer, we should be concerned about its effects. In the "Bold-Stroke" cockpit of the future for night all-weather missions , the pilot will look at a target and the computer will display a reticle indicating estimated eye position. There necessarily will be spatial offsets and temporal delays between the pilot's actual eye position and the position indicated by the computer. We should study the effects of these spatial offsets and temporal delays. We could do this by displaying a red target dot, and a white dot indicating the computer's estimate of the eye position. Then we could introduce a spatial offset, or a temporal delay, and ask the subject to superimpose the eye position reticle on the target dot. We could measure the mean square error between eye and target position, and plot it as a function of the magnitude of the temporal delay or the spatial offset. This type of a plot would show what tradeoffs can be made between spatial offset and temporal delay. This would let designers decide whether they want to invest more money in increasing the speed or the resolution of the eye movement monitors. We could also perform psychophysical experiments where the subject reports any detectable changes in target position or luminosity, or in spatial offset or temporal delay. This type of a study should allow us to estimate the instrumentation constraints that will allow successful visual processing in the cockpit of the future.

SUMMARY

The Honeywell oculometer has a noise level of about 0.1 deg^2: eye tracking is noisier than head tracking; vertical eye tracking is noisier than horizontal eye tracking. It has about 25% crosstalk of the horizontal channel into the vertical channel. It has an 84 msec time delay. It is not very effective at detecting and rejecting eye blinks; typical eye blink artifacts last 50 to 200 ms.

The human tracks best when tracking with eyes alone. Although tracking with head and eyes should be more natural, the human does worse when he uses his head. Head only tracking is the worst of the three conditions. In the tracking shown in this report the head only tracking had a mean squared error of 1.09 deg^2 and a time delay of 184 msec, the head and eye tracking had an error of 0.5 deg^2 and a delay of 150 msec, and the eyes only tracking had an error of 0.35 deg^2 and a delay of 84 msec.

REFERENCES

Bahill A.T., Kallman J.S. and Lieberman J.E. (1982) Frequency limitations of the two-point central difference differentiation algorithm. Biological. Cybernetics. 45, 1-5.

Bahill A.T. and LaRitz T. (1983) Do baseball and cricket players keep their eyes on the ball?. In Proc 1983 Int Conf Sys Man and Cybern (India) pp. 79-83. IEEE, New York.

Bahill A.T. and McDonald J.D. (1983) Smooth pursuit eye movements in response to predictable target motions. Vision. Res. 23, 1573-1583.

Barnes G.R. (1980) Vestibular control of oculomotor and postural mechanisms. Clin. Phys. Physiol. Meas. 1, 3-40.

Eizenman M., Frecker R.C. and Hallett P.E. (1984) Precise non-contacting measurement of eye movements using the corneal reflex. Vision. Res. 24, 167-174.

Hsu F.K., Bahill A.T. and Stark L. (1976) Parametric sensitivity of a homeomorphic model for saccadic and vergence eye movements. Comput. Prog. Biomedicine. 6, 108-116.

Merchant J., Morrissette R. and Porterfield J.L. (1974) Remote measurement of eye direction allowing subject motion over one cubic foot of space. IEEE. Trans. Biomed. Engr. BME-21, 309-317.

Morasso P., Bizzi E. and Dichgans J. (1973) Adjustment of saccade characteristics during head movements. Exp. Brain. Res. 16, 492-500.

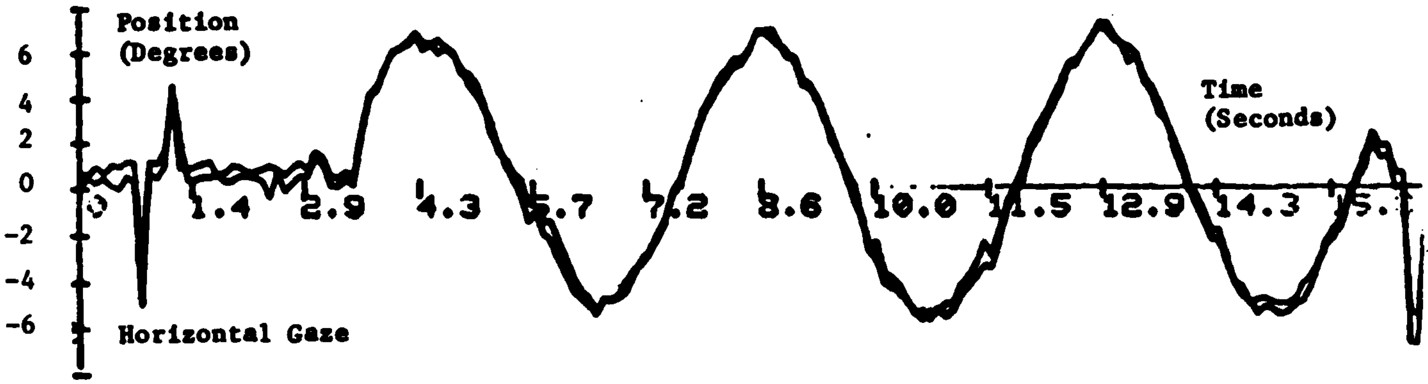


Figure 1. Horizontal gaze for coordinated eye and head tracking of a sinusoidally moving target. Gaze is the direction in which the subject is looking; it is the sum of eye position and head position. The gaze axis is labeled in degrees, with reference to straight ahead. The time axis is labeled in seconds. See text for a discussion of reliability, and an explanation of the two curves. Summary: When the target moves sinusoidally, the human's gaze also moves sinusoidally.

Fig. 2. Eyes only tracking of a target moving in a circle. The target started at the center, jumped 5 degrees up at 3 sec, made 3 clockwise rotations, and then jumped back to the center at 16 sec. For this trial, the subject was instructed to hold his head still. The traces are, from top to bottom, horizontal target position, vertical eye position, horizontal eye position, head position (roll), head position (pitch), head position (yaw), and horizontal gaze superimposed on the horizontal target position. Horizontal gaze is the sum of the horizontal eye position, and the horizontal head position (yaw).

Horizontal target position is the smooth curve in the bottom trace; it is the same as the top trace. Upward deflections are rightward or upward movements. Time axis is labeled in seconds. The position axes are labeled in degrees. Summary: The subject tracked well using only his eyes.

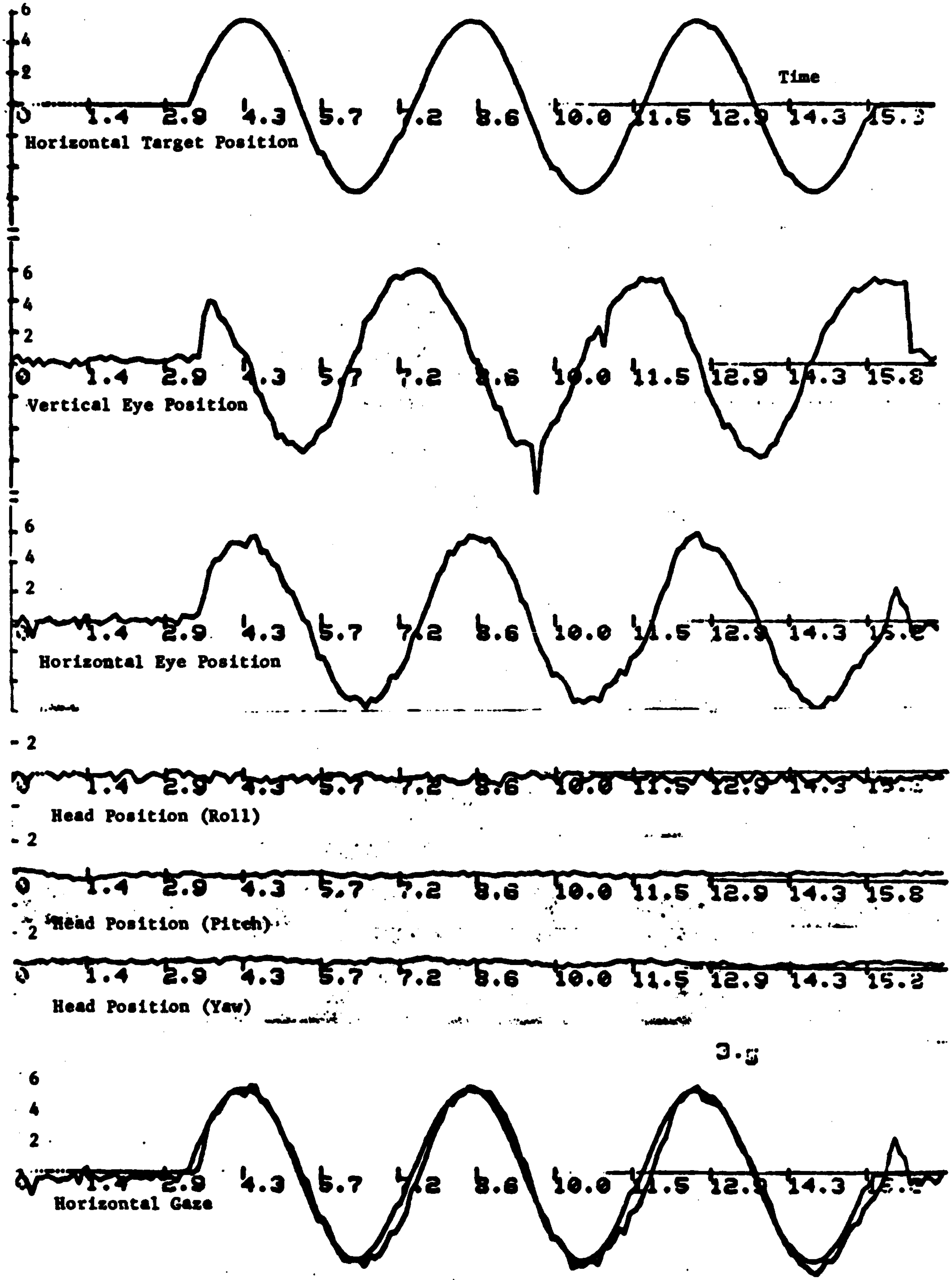

Time
Horizontal Target Position
Vertical Eye Position
Horizontal Eye Position
Head Position (Roll)
Head Position (Pitch)
Head Position (Yaw)
Horizontal Gaze

Figure 2

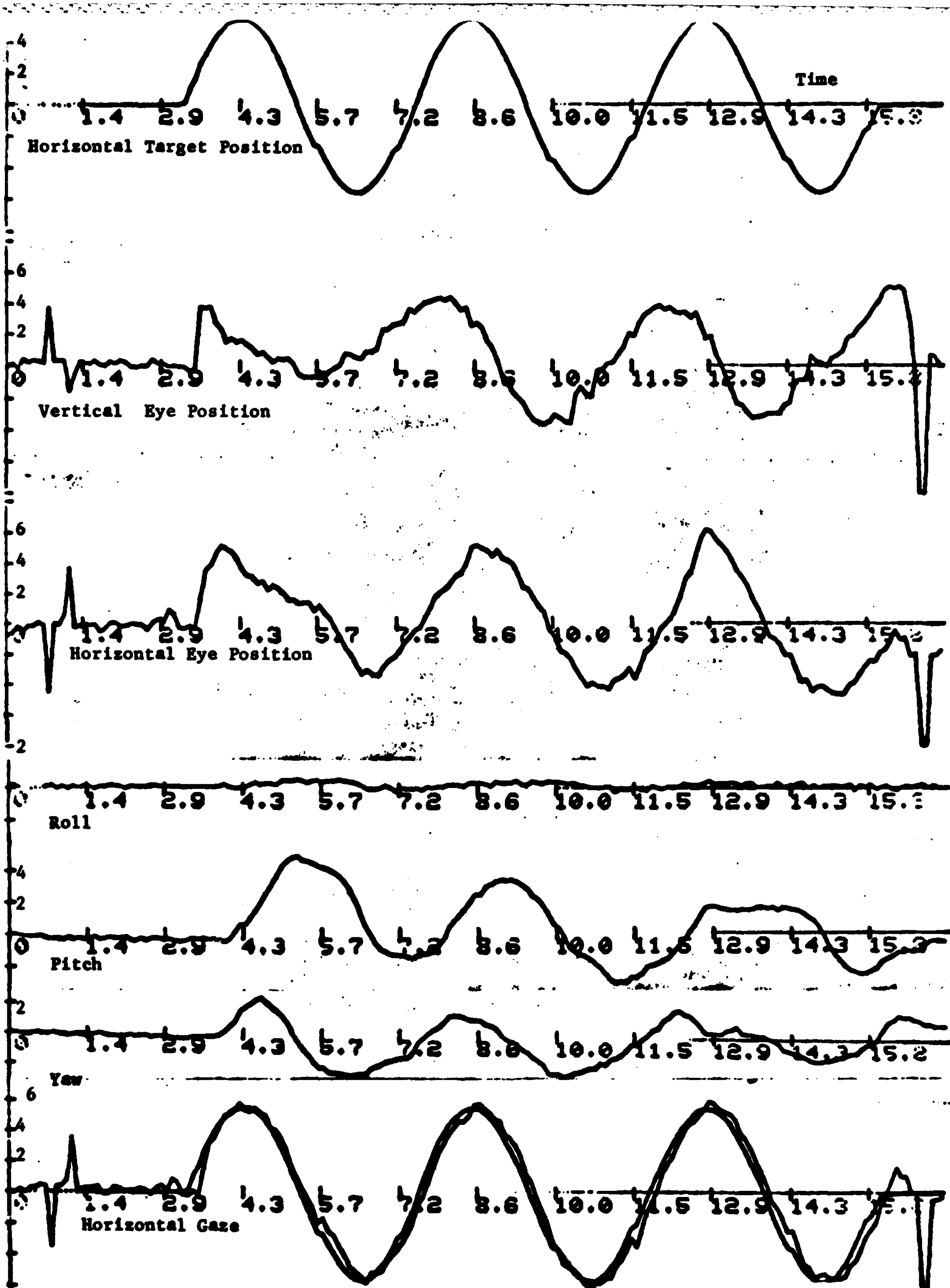


Fig. 3. Tracking with head and eyes. Same display format as in Fig. 2. Summary: The subject tracked almost as well using head and eyes.

Fig. 4. Head only tracking of a target moving in a circle. The subject was instructed to eliminate eye movements by superimposing a mark on the helmet visor (the image of the light filiment) on the target. Same display format as in Fig. 2. Summary: Horizontal gaze has the most error for head only tracking.

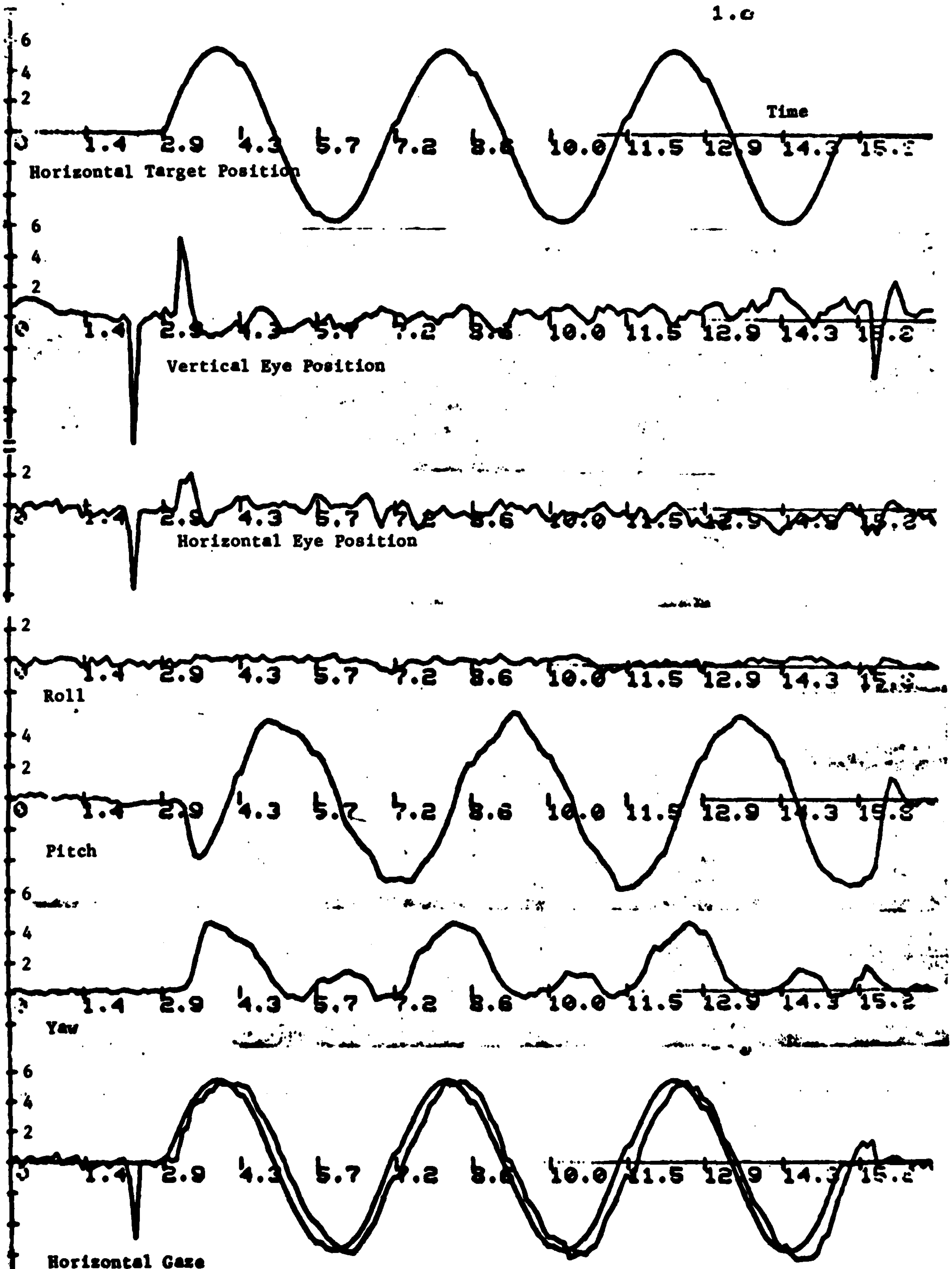

Time
Horizontal Target Position
Vertical Eye Position
Horizontal Eye Position
Roll
Pitch
Yaw
Horizontal Gaze

Figure 4

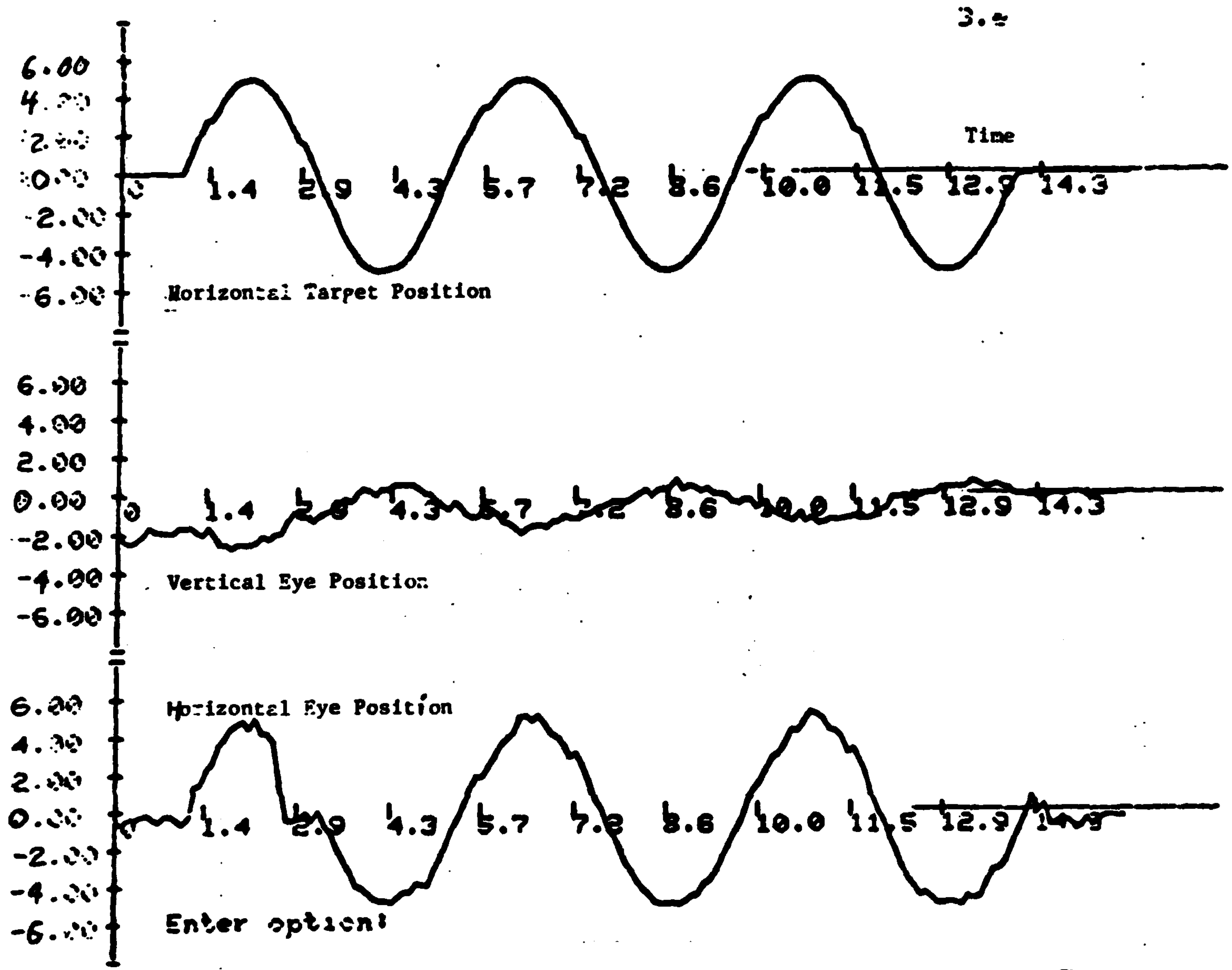


Figure 5. Eye only tracking of a pure horizontal target movement. Shown, from top to bottom, are horizontal target position, vertical eye position, and horizontal eye position. The apparent vertical eye movement is artifactual. Summary: There is crosstalk in the instrumentation system.

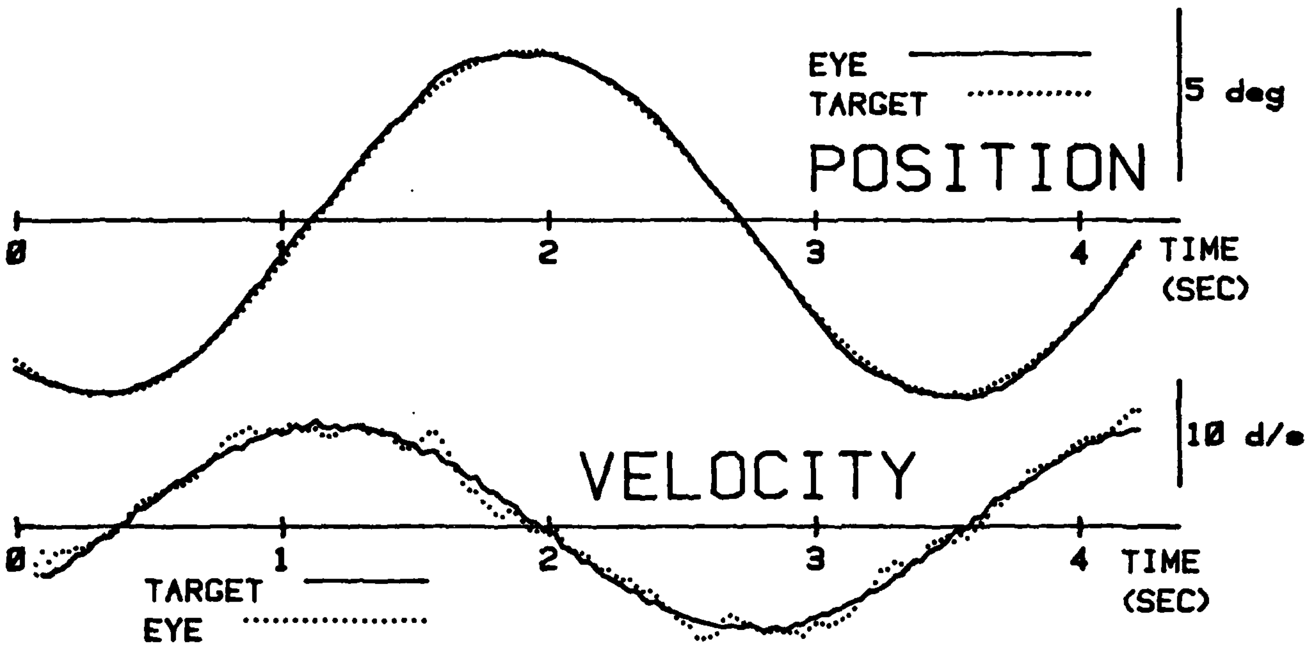


Fig. 6. Photoelectric data of an experienced subject tracking a sinusoid. Summary: There is no time delay.

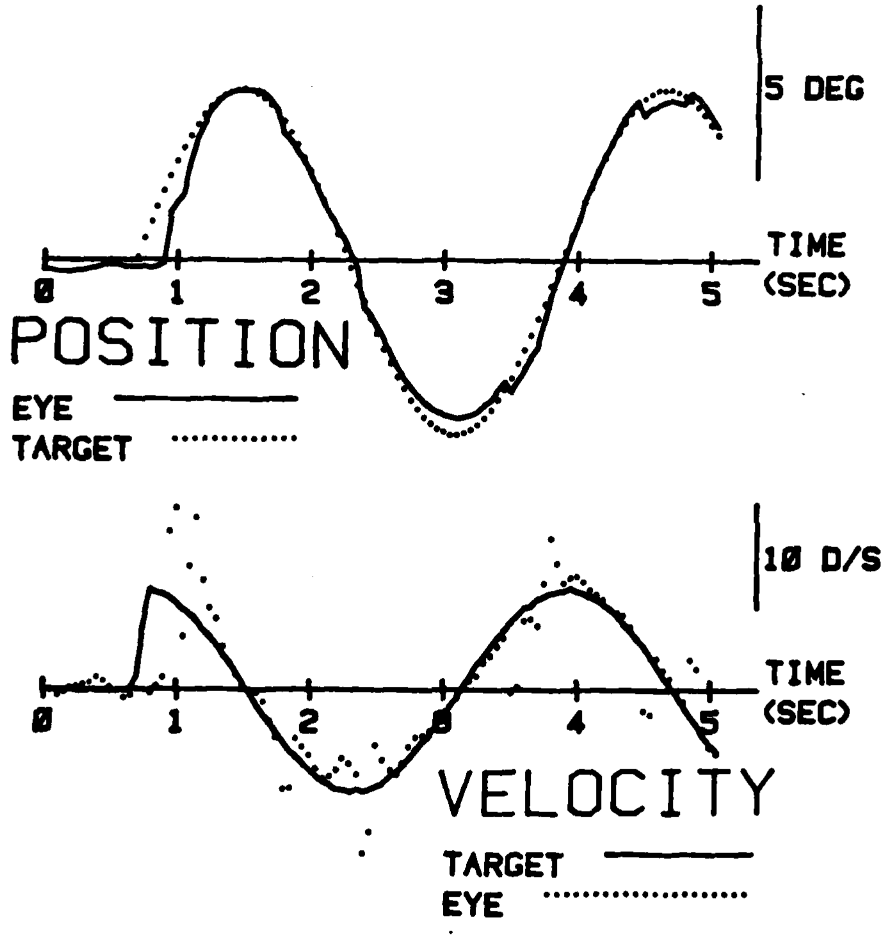


Fig. 7. Photoelectric data of a naive subject tracking a sinusoid. This is the first cycle of a sinusoid that he had seen in his life. Summary: There is no time delay.

APPENDIX

Mar 1 21:48 1984 hmof.c Page 1

```
#
/*********************************************************************
*                                                                    *
*       This program converts hmof data to standard NCSL three       *
*       channel data,                                                *
*       i.e. CHO = Horz eye, CH1 = Vert eye and CH2 = Horz target.   *
*       The original program was written by Jeff Kallman.            *
*                                                                    *
*       Last modified by Terry Bahill   February 1984                *
*                                                                    *
*       Compile with cc hmof.c -lS                                   *
*                                                                    *
*       The folllowing table shows the                               *
*       Data Transmitted from S/130 Eclipse Computer to DEC 11/34    *
*                  (16-bit Integers)                                 *
*  Word                                                              *
* Number          Name                         Contents              *
*                                                                    *
*    1       Adapter/Process           Hyperchannel I/O              *
*    2       Access Code               Control Words                 *
*    3       To Port                                                 *
*    4       From Port                                               *
*    5       Frame Counter             0 to 48 then recycle          *
*    6       STMDE                     Status/Mode                   *
*    7       HMO BITE                  HMO BITE Word                 *
*    8       HMS BITE                  HMS BITE Word                 *
*    9       A11                                                     *
*   10       A12                                                     *
*   11       A13                       Helmet direction cosine       *
*   12       A21                       rotational in Dome Coordinate *
*   13       A22                       System (DCS).                 *
*   14       A23                       (A12 represents matrix element *
*   15       A31                       of first row and second column.)*
*   16       A32                                                     *
*   17       A33                                                     *
*   18       EPX                                                     *
*   19       EPY                       Translation of tracked eye in *
*   20       EPZ                       DCS (inches)                  *
*   21       EVPX                                                    *
*   22       EVPY                      Translation of eye viewing point*
*   23       EVPZ                      in DCS (inches)               *
*   24       HMRX                                                    *
*   25       HMRY                      Translation of helmet magnetic *
*   26       HMRZ                      receiver in DCS (inches)      *
*   27       GX                        Tracked eye look vector       *
*   28       GY                        direction cosines in helmet   *
*   29       GZ                        coordinate system (HCS).      *
*   30       MFX                                                     *
*   31       MFY                       Tracked eye look vector       *
*   32       MFZ                       direction cosines in DCS      *
*   33       Horz Target (Az)                                        *
*   34       Vert Target (El)                                        *
*   35       Number 10,000 of a second between clock ticks           *
*   36       Number of clock ticks between interrupts (CKCNT)        *
*   37       Time since clock started:    seconds                    *
*   38                                    plus 0.0001 of a second    *
```

```
                                                                          *
  ***************************************************************************/

#define PI 3.14159              /* the value of pi */
struct cblk {                   /* structure of the calibration factor block */
                double leftfac;         /* the left eye factor */
                double rightfac;        /* the right eye factor */
                double tarsiz;          /* the size of the ideal eye jump */
                double tarfac;          /* the target factor */
                int leftoff;            /* the offset of the left eye */
                int rightoff;           /* the offset of the right eye */
                int taroff;             /* the offset of the target */
        } *factor;
        int lastleft, lastright, lasttarg, lp, lr, ly;
main()
{
        int inbuf[256], frame[43], eyebuf[770], hedbuf[770],gazbuf[770];
        int infildes, eyefildes, hedfildes,gazfildes;
        int lowlim, uplim, blockno, frameno, bytes, sofar;
        char filename[30], ans;
        lastleft= lastright= lasttarg= lp= lr= ly= 0;
        printf("What is the name of the input file? ");
        scanf("%s", filename);
        if((infildes= open(filename,0)) == -1)
        {
                printf("I'm sorry, I can't get to %s.  Try again? ",filename);
                scanf("%s",filename);
                if((infildes= open(filename,2)) == -1) exit(0);
        }
        printf("What do you want to name the output eye file?  ");
        scanf("%s", filename);
        if((eyefildes= open(filename,2)) != -1)
        {
                printf("File %s already exists. Should I overwrite? ",filename);
                scanf("%1s",&ans);
                if(ans == 'y') eyefildes= creat(filename,2);
                else
                {
                        printf("Please input new eye filename :");
                        scanf("%s", filename);
                }
        }
        else if((eyefildes= creat(filename,0644)) == -1)
        {
                printf("I'm sorry, I can't create %s. Try again? ",filename);
                scanf("%s", filename);
                if((eyefildes= creat(filename,0644)) == -1) exit(0);
        }
        printf("What do you want to name the output head file?  ");
        scanf("%s", filename);
        if((hedfildes= open(filename,2)) != -1)
        {
                printf("File %s already exists. Should I overwrite? ",filename);
                scanf("%1s",&ans);
                if(ans == 'y') hedfildes= creat(filename,2);
```

```
			else
			{
				printf("Please input new head filename :");
				scanf("%s", filename);
			}
		}
		else if((hedfildes= creat(filename,0644)) == -1)
		{
			printf("I'm sorry, I can't create %s. Try again? ",filename);
			scanf("%s", filename);
			if((hedfildes= creat(filename,0644)) == -1) exit(0);
		}
		printf("What do you want to name the output gaze file?  ");
		scanf("%s", filename);
		if((gazfildes= open(filename,2)) != -1)
		{
			printf("File %s already exists. Should I overwrite? ",filename);
			scanf("%1s",&ans);
			if(ans == 'y') gazfildes= creat(filename,2);
			else
			{
				printf("Please input new gaze filename :");
				scanf("%s", filename);
			}
		}
		else if((gazfildes= creat(filename,0644)) == -1)
		{
			printf("I'm sorry, I can't create %s. Try again? ",filename);
			scanf("%s", filename);
			if((gazfildes= creat(filename,0644)) == -1) exit(0);
		}
		printf("At which block should I start processing? ");
		scanf("%d", &lowlim);
		printf("At which block should I finish processing? ");
		scanf("%d", &uplim);
		if(uplim < lowlim) exit(0);
		printf("What is the peak to peak amplitude of target movement?");
		scanf("%f", &factor->tarsiz);
		printf("tarsiz is %f \n", factor->tarsiz);
/* Set nice gain and offset values  */
		factor->leftfac= 180.;
		factor->rightfac= 180.;
		factor->tarfac= 180.;
		factor->leftoff= 2048;
		factor->rightoff= 2048;
		factor->taroff= 2048;
/* Write calibration block of eye file. 38 bytes for 4 doubles and 3 integers */
		if((bytes= write(eyefildes,factor,38)) != 38) exit(0);
		for(bytes= 0; bytes <= 236; bytes++)
			inbuf[bytes]= 2048;
		if((bytes= write(eyefildes,inbuf,474)) != 474) exit(0);
/* Write calibration block of head file  */
		if((bytes= write(hedfildes,factor,38)) != 38) exit(0);
		for(bytes= 0; bytes <= 236; bytes++)
			inbuf[bytes]= 2048;
		if((bytes= write(hedfildes,inbuf,474)) != 474) exit(0);
```

```
/* Write calibration block of gaze file  */
        if((bytes= write(gazfildes,factor,38)) != 38) exit(0);
        for(bytes= 0; bytes <= 236; bytes++)
                inbuf[bytes]= 2048;
        if((bytes= write(gazfildes,inbuf,474)) != 474) exit(0);
        seek(infildes,lowlim,3);
        sofar= 0;
        for(blockno= lowlim; blockno <= uplim; blockno++)
        {
                if((blockno-lowlim)%100 == 0) printf("block= %d\n",blockno);
                if((bytes= read(infildes,inbuf,512)) != 512) exit(0);
                /*This stuff is here because we have 6 frames of 42 words
                  per block with four garbage words at the end of the block. */
                for(frameno= 0; frameno <= 5; frameno++)
                {
                        for(bytes= 0; bytes <= 41; bytes++)
                        {
                                frame[bytes + 1]= inbuf[frameno*42+bytes];
                                /*I wanted [bytes + 1] so that our unix numbers
                                would be the same as hmofs, namely 1 to 42 */
                        }
                        eyeframe(frame);
                        eyebuf[sofar]= frame[0];
                        eyebuf[sofar+1]= frame[1];
                        eyebuf[sofar+2]= frame[2];
                        headframe(frame);
                        hedbuf[sofar]= frame[0];
                        hedbuf[sofar+1]= frame[1];
                        hedbuf[sofar+2]= frame[2];
                        gazeframe(frame);
                        gazbuf[sofar]= frame[0];
                        gazbuf[sofar+1]= frame[1];
                        gazbuf[sofar+2]= frame[2];
                        sofar= sofar+3;
                        if(sofar > 766)
                        {
                        /* It writes out 3 blocks at a time */
                        /*3 output blocks come from 42 and 4/6 input blocks */
                        if((bytes= write(eyefildes,eyebuf,1536)) != 1536)
                                exit(0);
                        if((bytes= write(hedfildes,hedbuf,1536)) != 1536)
                                exit(0);
                        if((bytes= write(gazfildes,gazbuf,1536)) != 1536)
                                exit(0);
                        sofar= 0;
                        }
                }
        }
        printf("All done sir\n");
        exit(0);
}

gazeframe(buf)
/*This subroutine looks at all 42 words of frame[] as input,
and writes the first three words of buf[] for output.
It outputs AZ, and EL of gaze, and HMOF word 33, which is horz target  */
```

```
int buf[];
{
        double dcx, dcy, dcz, az, el;
        double asin(), atan2(), sqrt(), fabs();
                dcx= buf[30]/32767.;
                if(dcx >= 1.) dcx= .999;
                dcy= buf[31]/32767.;
                if(dcy >= 1.) dcy= .999;
                dcz= buf[32]/32767.;
                if(dcz >= 1.) dcz= .999;
        if(fabs(dcx) <= .0001) dcx= .0001;
        if(fabs(dcy) <= .0001) dcy= .0001;
        az= 180.*atan2(dcy,dcx)/PI;
        el= 180.*asin(-dcz)/PI;
        buf[0]= az*180+2048;
        buf[1]= el*180+2048;
        buf[2]= buf[33]+2048;
}

eyeframe(buf)
/*This subroutine looks at all 42 words of frame[] as input,
and loads up the first three words of buf[] for output.
It outputs horizontal and vertical eye position,
and HMOF word 33, which is horz target. */
int buf[];
{
        double dcx, dcy, dcz, az, el;
        double asin(), atan2(), sqrt(), fabs();
                dcx= buf[27]/32767.;
                if(dcx >= 1.) dcx= .999;
                dcy= buf[28]/32767.;
                if(dcy >= 1.) dcy= .999;
                dcz= buf[29]/32767.;
                if(dcz >= 1.) dcz= .999;
        if(fabs(dcx) <= .0001) dcx= .0001;
        if(fabs(dcy) <= .0001) dcy= .0001;
        az= 180.*atan2(dcy,dcx)/PI;
        el= 180.*asin(-dcz)/PI;
        buf[0]= az*180+2048;
        buf[1]= el*180+2048;
        buf[2]= buf[33]+2048;
}

headframe(buf)
/*This subroutine looks at all 42 words of frame[] as input,
and loads up the first three words of buf[] for output.
It outputs yaw, pitch, and roll.   */
int buf[];
{
        double yaw, pitch, roll, a31, a11, a33, cosphi;
        double asin(), acos(), cos(), fabs();
                a31= buf[15]/32767.;
                a11= buf[9]/32767.;
                a33= buf[17]/32767.;
        pitch= 180.*asin(-a31)/PI;
```

```
        cosphi= cos(pitch*PI/180.);
        if(fabs(cosphi) <= .0001) cosphi= .0001;
        yaw= 180.*acos(a11/cosphi)/PI;
        roll= 180.*acos(a33/cosphi)/PI;
        buf[0]= yaw*180.+2048;
        buf[1]= pitch*180.+2048;
        buf[2]= roll*180.+2048;
}

double acos(x)
        double x;
        {
        double y, z;
        double atan(),sqrt(),fabs();
        if(fabs(x) >= .9999) x= .9999;
        if(fabs(x) <= .0001) x= .0001;
        y= sqrt(1.-x*x);
        z= atan(y/x);
        if (x<0) z= PI + z;
        return(z);
        }

double asin(y)
        double y;
        {
        double x, z;
        double atan(),sqrt(),fabs();
        if(fabs(y) >= .9999) y= .9999;
        if(fabs(y) <= .0001) y= .0001;
        x= sqrt(1.-y*y);
        z= atan(y/x);
        return(z);
        }
```